Usage:

Model Training:

python3 train <examples> <features> <hypothesisOut> <learning-type>

- **examples** is a file containing labeled (“en” for English; “nl” for Dutch) examples in the format label|examples
- **features** is a file containing the features used to train. File is formatted where every feature corresponds to a line.
- **hypothesisOut** specifies the filename to write your model to.
- **learning-type** specifies the type of learning algorithm to run ("dt" or "ada").

Model Prediction:

predict <examples> <features> <hypothesis>

- **examples** is a file containing lines of 15 word sentence fragments in either English or Dutch.
- **features** is a file containing the features used to train the hypothesis.
- **hypothesis** is a trained decision tree or ensemble created by your train program

Example Usage:

test.dat:

Hello, this is English
Hallo, dit is nederlands

input:

python3 lab3.py train data.txt features.txt best.model ada
python3 lab3.py predict test.dat features.txt best.model

output:

en
nl

Features Description:

- Features used in the model are common words,character, and bigrams in English and Dutch. These features were chosen based on their frequency and distinctiveness from websites such as:

    - https://www3.nd.edu/~busiforc/handouts/cryptography/Letter%20Frequencies.html
    - https://www.sttmedia.com/syllablefrequency-dutch
- The features are:
    - **English Features:** "the", "a", "to", "be", "of", "and", "y", "is", "that", "have", "it", "for", "not", "on", "am", "I"
    - **Dutch Features:** "de", "een", "het", "met", "van", "te", "dat", "die", "zijn", "op", "ui", "oe"

Decision Tree Description:

- The decision tree is built using the LEARN_DECISION_TREE function based on the book's pseudocode.
- The function recursively splits the data based on the attribute with the highest information gain. The tree stops growing when all examples have the same classification, no attributes are left, or when the max depth is reached in case of AdaBoost.
- Parameters:
    - The maximum depth for the decision tree is set to 1 when used with AdaBoost to ensure weak learners.
    - The gain is calculated using the weighted entropy function, the weight is set to 1 for the Decision Tree algorithm.

AdaBoost Description:

- The AdaBoost algorithm is implemented in the AdaBoost class.
- It iteratively trains weak decision tree classifiers and combines them into a strong classifier.
- The weights of the examples are updated based on the errors of the weak classifiers.
- Number of Trees:
    - The number of stumps used in AdaBoost is set to 8.
    - This number was chosen based on the training phase stopping at stumps number 8 due to error > 0.5.